SPECIFICATION AMENDMENTS

Substitute the paragraph [0033] at page 7, with the following:

[0033] It is clear from the illustration in Figure 1 that supporting users of multiple versions of the illustrated database can be very cumbersome as the users upgrade from one version to another. For example, when version 1.2 is released, new users will install version 1.2 of the database from scratch. In this scenario, an installation file for installing version 1.2 of the relational database accounts for database objects created in version 1.0, version 1.1, and version 1.2, as well as any modifications to objects performed when upgrading to version 1.1 or version 1.2, as well as any deletes performed when upgrading to version 1.1 or ~~version 1.2are~~ version 1.2 are included in.

Substitute the paragraph [0049] beginning on page 12, with the following:

[0049] As indicated in Figure 4, the union of set 402 and set 404 results in a version 1.0 installation file 406 for the relational database.

Substitute the paragraph [0067] beginning on page 18, with the following:

[0067] Figure 9 illustrates select components of an exemplary DSVMS 814, illustrated within an exemplary software development environment 900. DSVMS 814 includes database schema version management XML schema definition (DSVM XSD) 902, DSVM XML file 904, data definition language (DDL) scripts 906, data manipulation language (DML) scripts 908, and upgrade/installation file ~~generator 910~~ generator 910.

Substitute the paragraph [0070] beginning on page 19, with the following:

[0070] Upgrade/installation file generator 910 is configured to assemble a file of scripts that, when executed, will install a particular version of a relational database, or upgrade an existing installation of the database from one version to another. In an exemplary implementation, upgrade/installation file generator 910 applies laws of set theory (e.g., as described above with reference to Figures 4-7) to database schema metadata stored in DSVM XML file 904. The results of the set theory calculations are then used to identify specific DDL scripts 906 and DML scripts 908 that are to be included in a particular upgrade or installation file. In an exemplary implementation, upgrade/installation file generator 910 is implemented using the C# programming language. However, it is recognized

that any number of programming languages may be used to implement upgrade/installation file generator 910 including, but not limited to, the C programming language, the C++ programming language, or ~~Java~~ the JAVA programming language.

Substitute the paragraph [0083] beginning on page 23, with the following:

[0083] In an alternate implementation, rather than organizing the scripts by subdirectory, the scripts may be managed using a source version control system (e.g., ~~ClearCase, Source Safe,~~ CLEARCASE, SOURCE SAFE, CVS, etc.). The source version control system may maintain multiple versions of a single script file, as the file is modified through multiple versions of the relational database. In this implementation, the XML file in which the database schema data is stored may also include a version number associated with each script file such that the version number corresponds to a version number maintained by the source version control system. This type of an implementation enables the creation of an installation file for any version of the relational database (current or previous), as well as the creation of an upgrade file from any previous version to any newer version.